



Informatica senza e con computer nella Scuola Primaria

Michael Lodi, Renzo Davoli, Rebecca Montanari, Simone Martini

► To cite this version:

Michael Lodi, Renzo Davoli, Rebecca Montanari, Simone Martini. Informatica senza e con computer nella Scuola Primaria. Enrico Nardelli. Coding e oltre: l'Informatica nella scuola, , 2020. hal-02379212

HAL Id: hal-02379212

<https://inria.hal.science/hal-02379212>

Submitted on 25 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Informatica senza e con computer nella Scuola Primaria¹

Michael Lodi*, Renzo Davoli, Rebecca Montanari, Simone Martini*

Dipartimento di Informatica - Scienza e Ingegneria (DISI)

Alma Mater Studiorum - Università di Bologna

**anche INRIA Focus team, Sophia-Antipolis, France*

Abstract

Questo capitolo descrive le esperienze svolte da docenti e ricercatori del Dipartimento di Informatica - Scienza e Ingegneria (DISI) dell'Alma Mater Studiorum - Università di Bologna nell'ambito di alcuni progetti di disseminazione dei principi di base dell'Informatica (pensiero computazionale) nelle scuole primarie bolognesi.

L'approccio che caratterizza i progetti presentati è l'utilizzo, sequenziale o intervallato, di attività sia "unplugged" sia "plugged". Le prime - svolte senza l'utilizzo di un calcolatore, ma con materiali e oggetti di uso comune - sono state in parte tratte dal libro CS Unplugged, e in parte ideate dagli autori; le seconde hanno fatto uso del linguaggio visuale a blocchi Scratch. Con riferimento alla Proposta di Indicazioni Nazionali per l'Informatica nella Scuola [CINI17], tali attività hanno coperto soprattutto gli ambiti prettamente informatici di rappresentazione dell'informazione (es. numeri binari), algoritmi (es. massimo) e loro descrizione in uno pseudolinguaggio, programmazione; ma non hanno tralasciato quelli di creatività digitale (realizzare un progetto personale con Scratch) e consapevolezza digitale (es. regole di comportamento sulla community online di Scratch).

Introduzione

Come docenti e ricercatori di Informatica, siamo convinti che faccia parte della nostra missione quella di condividere e diffondere i principi base della nostra disciplina: tanto importante e radicata nella società di oggi (e a maggior ragione di domani), eppure tanto assente dalla quasi totalità dei percorsi della scuola dell'obbligo.

Abbiamo quindi ritenuto importante dedicare parte del nostro tempo a sperimentare attività di introduzione al "pensiero computazionale", espressione che usiamo come sinonimo di "principi base dell'Informatica": quelle *grandi idee* che caratterizzano la nostra bella disciplina e che dovrebbero essere patrimonio di ogni cittadino - non necessariamente perché diventi un professionista - ma perché tali principi possono essergli utili in qualsiasi ambito dello studio, del lavoro e della vita (così come accade per tutte le altre discipline insegnate a scuola).

Riteniamo inoltre questa una missione "a lungo termine" - intrapresa nella speranza di avere iscritti ai nostri corsi di Laurea che abbiano compiuto una scelta più consapevole e che posseggano già gli strumenti intellettuali di base per approfondire in maniera piena la

¹ This is an authors' pre-print version of the work. It is posted here for your personal use. Not for redistribution. The definitive version will be published as a chapter of the book "Coding e oltre: Informatica nella scuola", edited by Enrico Nardelli and published by Lisciani in 2019.

disciplina.

Abbiamo deciso di partire dalla scuola primaria, dove i bambini sono totalmente aperti e curiosi di scoprire il mondo e non sono ancora influenzati da stereotipi (ad esempio che l'Informatica sia un'attività prettamente maschile).

La struttura del percorso

Si è deciso di proporre attività pratiche, in cui gli studenti siano attivi e partecipi in prima persona: le attività sono quindi ispirate, dove possibile, all'apprendimento per scoperta e ad approcci di tipo costruttivista e costruzionista.

I bambini vengono posti in ambienti progettati per permettere loro l'esplorazione e la costruzione delle proprie conoscenze, indirizzati però verso gli obiettivi di apprendimento che verranno dettagliati nella descrizione delle diverse proposte.

Quelle che presenteremo sono una serie di attività, alcune autocontenute e dunque esauribili in una o due lezioni, altre strutturate in percorsi di più lezioni.

Il filo conduttore è la dialettica tra attività "unplugged", cioè svolte senza l'uso del computer, sia "plugged", cioè svolte tramite ambienti di programmazione adatti.

L'iperesposizione degli alunni alla tecnologia fino dai primi anni di vita li rende molto abili nell'uso di strumenti anche complessi come telefoni cellulari, tablet, smart TV, e così via.

È una conoscenza operativa volta ad ottenere il risultato voluto, spesso appresa per tentativi o per passaparola. Sono sicuramente capacità utili nella vita di tutti i giorni del XXI secolo ma possono anche diventare limitanti se non accompagnate ad una conoscenza più profonda dei meccanismi e dei metodi che fanno funzionare gli strumenti tecnologici.

Come insegnanti occorre trovare metodologie didattiche in grado di generare l'interesse dei giovani discenti. Lo studio del pensiero computazionale senza computer mediante giochi in aula o in cortile, con modalità *unplugged*², consente:

- di non usare alcuna formalizzazione per la presentazione dei concetti,
- di allenare la fantasia per creare modelli di funzionamento degli strumenti,
- di rompere l'erroneo stereotipo che porta a identificare *digitale* con *elettronico*,
- di consentire ad alunni con diverse esperienze di confrontarsi ad armi pari senza privilegiare coloro che hanno maggior disponibilità di gadget tecnologici e quindi maggior dimestichezza.

La ricerca [BV18] svolta in questi anni sulle attività unplugged ha mostrato che esse risultano significative soprattutto quando i concetti appresi con tali attività vengono poi riutilizzati e "concretizzati" tramite l'utilizzo di strumenti di programmazione su un elaboratore.

Di più: studenti che hanno svolto attività unplugged sembrano utilizzare in maniera più

² Tim Bell (University of Canterbury, Nuova Zelanda) ha proposto, già da decenni, alcune attività per insegnare concetti di Informatica tramite giochi che non richiedono l'uso di un computer. Per una rassegna sull'evoluzione del concetto, si veda [BV18].

La versione attuale (<https://csunplugged.org/>) è una raccolta di attività (con materiali scaricabili e adattabili direttamente nel sito) e percorsi pensati per l'integrazione delle attività unplugged in un curriculum scolastico. Il sito è attualmente in inglese, tedesco e spagnolo. Una versione precedente del progetto è costituita da un libro liberamente scaricabile. La traduzione italiana del libro, a cura di Renzo Davoli, Giovanni Michele Bianco, Piergiovanna Grossi, è disponibile a: <https://classic.csunplugged.org/books/>.

completa i linguaggi a blocchi (rispetto ai compagni che hanno svolto solo attività “plugged”) e avere un maggiore senso di autoefficacia nei confronti della disciplina.

Come diremo nelle conclusioni, anche se solo aneddoticamente, abbiamo osservato analoghi risultati durante i nostri interventi.

Le attività

Attività senza computer

Forti della lunga esperienza internazionale sulle metodologie *unplugged* abbiamo elaborato alcune proposte:

- i numeri binari con le monete
- la trasmissione dei caratteri con le luci (o i cartellini colorati)
- la ricerca del massimo con le bilance.

Nel seguito di questo capitolo verranno illustrate queste attività anche per mostrare come CS Unplugged sia da considerare una metodologia didattica e non solamente un interessante libro/raccolta di attività.

Infatti - come si potrà rilevare dalla descrizione che segue - ci si è ispirati ad attività presentate nel libro in modo creativo, proponendo lezioni e usando strumenti/materiali/oggetti fisici diversi.

I materiali sono liberamente consultabili, stampabili e adattabili³.

Le esperienze sono state svolte nella stessa classe in tre anni diversi, in seconda, terza e quarta primaria.

[inizio parte eventualmente rimovibile]

Le stesse attività sono state poi inserite in un percorso di Alternanza Scuola Lavoro con studenti di un Liceo Scientifico di Bologna. Gli studenti del Liceo "Fermi" hanno partecipato ad un corso di formazione in Università sulla didattica del pensiero computazionale e poi come attività simil-lavorativa hanno svolto funzioni di mentor per insegnanti delle scuole primarie e secondarie di primo grado dell'istituto comprensivo di Ceretolo a Casalecchio di Reno a pochi chilometri dalla città.

In breve l'esperienza di alternanza scuola-lavoro:

- è una esperienza specifica per studenti del Liceo che hanno come possibile prospettiva occupazionale, sebbene a lungo termine, quella dell'insegnamento,
- consente agli studenti di misurarsi con un apprendimento volto non già alla propria maturazione culturale (o peggio al superamento di una verifica o interrogazione) ma alla trasmissione del sapere,
- responsabilizza gli studenti liceali che hanno dato prova di prendere molto seriamente l'impegno,
- consente agli insegnanti delle scuole di cimentarsi in una collaborazione con mentor più giovani che hanno la dimestichezza tecnologica tipica dei nativi digitali,
- evita il contatto diretto fra studenti liceali e gli alunni: l'interazione in una reale classe è una attività delicata, occorre la professionalità di operatori con conoscenze di pedagogia, l'interazione diretta con adolescenti richiederebbe uno studio specifico e una puntuale supervisione.

[fine parte eventualmente rimovibile]

I numeri binari con le monete

Ogni buona lezione nasce dalla fantasia, e nulla può scatenare la fantasia quanto una favola. *"Ci sono due mondi: il mondo degli oggetti fatti di materia (Materialia) e quello di ciò che è costruito di conoscenza (Logicalia). In un paese del mondo di Logicalia che potremmo chiamare Digitalia Binaria esistono due sole cifre: lo zero e l'uno. Gli abitanti di Logicalia Binaria usano delle monete. Le monete hanno diversi colori e valori. Hanno una sola regola: non si possono mai usare due monete dello stesso valore per fare un pagamento... anche perché non conoscono il numero 2. Le monete valgono uno⁴, due, quattro, otto, sedici e trentadue bin.*

Gli abitanti possono pagare ogni importo da uno a sessantatre bin (non ci sono per ora cose che costano più di sessantatre bin in Digitalia Binaria, se dovesse essere necessario in futuro la zecca potrebbe coniare monete più preziose da sessantaquattro o centoventotto bin, ogni nuova moneta avrebbe il valore doppio rispetto alla precedente)."

Ad ogni alunno vengono date monete di colore diverso, una per tipo. Sul retro di ogni moneta c'è il valore riportato con la scritta "vale 1 x 1bin, 1 x 10bin, 1 x 100bin"... sul verso della moneta è stampato un numero di grossi punti colorati pari al valore della moneta.

Come primo esercizio, si invitano gli alunni ad usare le monete dal lato dei punti. Si generano casualmente (ma non troppo) numeri da zero a sessantatre e si invita a cercare quali monete servono per pagare quel valore. Ovviamente si inizia con numeri bassi, poi si continua con numeri che hanno comunque necessità di poche monete per essere rappresentati, poi si incrementa man mano la complessità.

Agli alunni viene dato anche un pannello con le sagome vuote delle monete con i colori in ordine dal più al meno significativo.

La discussione che segue porta a spiegare come quelli che consideriamo numeri sono in realtà i nomi dei numeri... I numeri hanno diversi nomi in diverse lingue, nella lingua e nella tradizione di Digitalia Binaria i numeri si chiamano con il numero di monete di ogni valore che è necessario mettere sul pannello per rappresentare il valore. Per esempio per rappresentare dodici occorrono 0 monete da trentadue, 0 da sedici, 1 da otto, 1 da quattro, 0 da due e 0 da uno, quindi 001100. Così se nella nostra tradizione siamo abituati a usare la numerazione decimale e ci sono le unità, le decine, le centinaia, migliaia e oltre, a Digitalia Binaria ci sono le unità, le duine, le quartine, le ottine, le sedicine, le trentadue e così via.

A questo punto il ghiaccio è rotto e si possono iniziare a fare operazioni coi numeri binari o a contare fino a trentadue con le dita di una mano o a milleventiquattro con le dita di entrambe le mani.

Il gioco con le dita suscita normalmente grande interesse (e puerile ilarità sui numeri 4 e 19).

A fine lezione occorre associare questo concetto con il funzionamento degli elaboratori: occorre progettare circuiti in grado di fare operazioni matematiche o di ricordare numeri. È possibile usare solo le cifre 0 e 1 per rappresentare ogni numero, perché fare macchine complesse quando si possono costruire semplici? E la tabellina in binario è molto più semplice di quella decimale... molti alunni della scuola primaria preferirebbero dover imparare a memoria la sola tabellina con numeri binari invece dell'intera tavola pitagorica!

Questa attività può contribuire a raggiungere il traguardo T-P-6 della Proposta di indicazioni nazionali [CINI17], mostra infatti come i numeri possano essere rappresentati dagli elaboratori per una più efficiente elaborazione.

⁴ i numeri in notazione decimale vengono riportati in parola, come occorre fare durante la lezione, 100 deve essere pronunciato uno-zero-zero

È importante sottolineare come questa attività contribuisca anche a raggiungere obiettivi di Matematica per la primaria nell'ambito "Numeri", così come da Indicazioni Nazionali⁵.

La trasmissione dei caratteri con le luci (o i cartellini colorati)

In Digitalia Binaria non solo i numeri si rappresentano solo con 0 e 1 ma anche le parole, le immagini, i suoni e forse anche i gusti e gli aromi.

Come potremmo scrivere o comunicare "ciao" se non abbiamo le lettere? È semplice: basta associare un numero binario ad ogni carattere alfabetico.

Per questa attività didattica è stato usato un pannello con otto led bicolore allineati che possono essere accesi con il colore rosso o verde. Per accendere i led si usano nove levette: una per accendere o spegnere l'intero pannello e le altre otto per decidere se ogni led verrà illuminato di rosso o verde.

Ad ogni gruppo di due studenti viene distribuita una tabella che riporta un estratto della codifica ASCII⁶ comprendente le lettere alfabetiche maiuscole, i numeri, lo spazio ed alcuni elementi di punteggiatura. La rappresentazione ASCII di ogni carattere è espressa come numero binario: ogni zero è stampato in colore verde e ogni uno in rosso.

Ora si invita ogni gruppo a inventare una parola non troppo lunga ma difficile da indovinare. A questo punto si invita un primo gruppo a "trasmettere", carattere dopo carattere, la propria parola. Appena un gruppo indovina la parola in corso di trasmissione viene invitato a trasmettere la propria. Si instaura così una gara, nella quale l'insegnante avrà l'accortezza di formare opportunamente i gruppi e dirigere lo svolgimento in modo che nessuno si senta escluso.

Gli alunni in questo modo apprendono gioiosamente e giocosamente che i numeri (in particolare quelli in notazione binaria) possono essere usati per rappresentare la conoscenza normalmente rappresentata con alfabeti diversi. Gli *access point* delle reti wireless o le stazioni radio base per i cellulari non sono poi tanto diversi dal nostro pannello con led bicolore.

Questa attività può contribuire al raggiungimento del traguardo T-P-6 come definito nella Proposta di indicazioni nazionali [CINI17]. Gli obiettivi dell'attività si collocano nell'*ambito dati e informazione*. Gli alunni comprendono come dati e informazioni possono essere codificati (O-P3-D-1) e come, con una legenda, possano essere decodificati (O-P3-D-2).

La ricerca del massimo con le bilance

Se la rappresentazione della conoscenza è un caposaldo del pensiero computazionale, il secondo e forse più importante elemento di base è l'elaborazione. Un percorso di base di avvicinamento al pensiero computazionale non poteva prescindere da una attività correlata alla elaborazione.

Il problema affrontato nella progettazione di questa attività è stato di coniugare la soluzione di un problema reale con la semplicità espositiva necessaria per l'età degli alunni. È stato scelto l'algoritmo del massimo perché consente di essere risolto senza dover introdurre concetti legati alle collezioni ordinate (e.g. vettori).

⁵ <http://www.indicazioninazionali.it/>

⁶ American Standard Code for Information Interchange, Codice Standard Americano per lo Scambio di Informazioni). Si tratta appunto di un sistema - ancora in uso - di codifica che associa ai caratteri usati in inglese un numero binario (originariamente a 7 bit: dunque 128 caratteri rappresentabili).

La strumentazione fornita ad ogni gruppo di sei/sette alunni è composta da:

- una bilancia a piatti contrapposti (autocostruita con righe di legno e piatti di plastica da pic-nic),
- otto barattoli di peso diverso (bicchierini da caffè con coperchio - in uso normalmente nei bar - riempiti di sale in diversa quantità),
- tre contenitori per i barattoli che abbiamo chiamato A, B e S,
- strisce di carta con istruzioni.

Scopo dell'esercizio è di trovare il barattolo di peso massimo seguendo una sequenza di istruzioni costruita ponendo sul banco una dopo l'altra alcune fra le strisce fornite.

Inizialmente tutti i barattoli sono nel contenitore A, il contenitore B è vuoto e il contenitore S alla fine deve contenere il solo barattolo più pesante, che costituisce la soluzione del problema.

Le istruzioni sono del tipo:

- prendi un barattolo dal contenitore A e mettilo su un piatto vuoto della bilancia
- prendi il barattolo che si trova nel piatto più in alto e mettilo nel contenitore B
- prendi il barattolo che si trova nel piatto più in basso e mettilo nel contenitore B
- prendi il barattolo che si trova nel piatto più in alto e mettilo nel contenitore S
- prendi il barattolo che si trova nel piatto più in basso e mettilo nel contenitore S

Si inizia con due barattoli, poi man mano si prova con più barattoli.

Ogni proposta di soluzione viene sperimentata all'interno del gruppo, eleggendo un componente a "robot" e un secondo a lettore dell'algoritmo. Occorre sorvegliare che il "robot" non prenda iniziative autonome e esegua in modo fedele ciò che viene richiesto.

Purtroppo con le sole istruzioni indicate occorre un programma distinto specifico per un certo numero di barattoli.

A questo punto si aggiunge l'istruzione "finché ci sono ancora barattoli nel contenitore A", (che viene disegnato in modo da realizzare il concetto di ciclo while) e si chiede di creare un algoritmo capace di funzionare per un numero arbitrario di barattoli.

Lo scopo di questa attività non è solamente quello di scrivere un algoritmo ma anche di trasformarlo in programma. Questo secondo passaggio infatti consiste nel trasformare il processo logico non ambiguo composto di un numero finito di passi elementari in sequenza di istruzioni di un linguaggio. Occorre saper scegliere le istruzioni appropriate, non stupirsi quando si scopre che esistono molteplici soluzioni corrette, saper valutare i limiti delle proprie implementazioni ed infine interrogarsi sul livello di generalità della soluzione proposta.

Con riferimento alla proposta di indicazioni nazionali [CINI17] questa attività vuole contribuire a raggiungere i traguardi T-P-1 (descrizione di una procedura con un algoritmo) e T-P-4 (verificare se il *programma* raggiunge il risultato voluto). In particolare gli obiettivi perseguiti si possono collocare nell'ambito algoritmi (O-P5-A-1, spiegare semplici algoritmi) e nell'ambito programmazione⁷ (O-P5-P-1 lettura e debug, O-P5-P-5 scelte a due vie, O-P5-P-2 cicli, O-P5-P-3 definizione di blocchi per la selezione o i cicli).

Domo: un maggiordomo "computazionale"

Oltre alle attività ispirate a CS Unplugged, è stata proposta un'attività legata in modo più specifico alla programmazione.

⁷ Sebbene si tratti di un'attività Unplugged, gli studenti devono tradurre il proprio ragionamento in un algoritmo e poi esprimerlo con un linguaggio formale (le istruzioni fornite su carta) che un esecutore (il bambino che ricopre il ruolo del "robot") dovrà interpretare.

Si tratta di una serie di lezioni progettata interamente da tre degli autori di questo capitolo. Il percorso è liberamente consultabile e modificabile [DLM17a].

Il percorso è pensato per introdurre i concetti di base della programmazione strutturata e alcuni semplici algoritmi, utilizzando attività *unplugged*.

Il percorso è stato pensato per una classe terza primaria che non abbia svolto precedentemente attività legate al pensiero computazionale, ma può essere adattato a ogni classe della scuola primaria, a patto che i bambini abbiano una buona capacità di lettura.

Il percorso è suddiviso in cinque lezioni da circa tre ore l'una.

Le attività sono pensate perché i bambini siano divisi in gruppi e ogni gruppo sia supportato da un facilitatore formato sugli argomenti del percorso.

Ogni lezione è strutturata con:

- un breve racconto introduttivo, che pone i personaggi in una specifica situazione problematica da risolvere; il testo può essere letto/raccontato agli studenti e/o stampato e fornito loro;
- del materiale da stampare e ritagliare e fornire agli studenti [DLM17a, DLM17b];
- una guida per l'insegnante.

Il contesto è quello di un personaggio che deve comandare, tramite delle istruzioni opportunamente codificate, un robot (il maggiordomo Domo) per muoversi su una griglia e compiere determinate azioni (es. raccogliere oggetti e portarli in punti specifici, uscire da un labirinto). La griglia può essere stampata su un foglio e Domo rappresentato con una pedina, oppure si può realizzare la griglia sul pavimento della classe e far impersonare Domo a uno o più studenti.

Tale approccio è condiviso da altre proposte didattiche, sia *unplugged* che *plugged* (es. CodyRoby, BeeBot, "Il labirinto" di Code.org, LightBot). Le attività qui descritte hanno similitudini con esse, ma presentano anche alcune peculiarità:

- durante la prima lezione, sono gli studenti stessi a dover scoprire le istruzioni comprese dal robot e a codificarle con una serie di simboli a loro scelta, scoprendo da un lato che i computer comprendono un set finito di istruzioni (es. il robot comprenderà "vai avanti di una casella" ma non "vai avanti") e dall'altro che i modi con cui rappresentiamo queste istruzioni sono solo una convenzione tra programmatore e computer, purché non ci sia ambiguità sulla loro semantica;
- le carte utilizzate "simulano fisicamente" il concetto di strutture di controllo: la sequenza è indicata semplicemente impilando le carte, mentre la selezione e l'iterazione, che possono prevedere nel loro "corpo" una o più istruzioni sono realizzate con delle tasche al cui interno è possibile inserire altre carte; questo introduce un'importante metafora fisica che aiuterà poi i bambini nella programmazione strutturata;
- man mano che si avanza nelle lezioni, agli studenti viene chiesto di scrivere programmi che risolvano non solo la specifica situazione della griglia, ma siano una soluzione generale a una classe di problemi simili (es. uscire da un labirinto qualsiasi, con certi vincoli); i bambini saranno incentivati a provare le proprie soluzioni, per esempio, su percorsi di altri gruppi (differenti dal proprio).

Le attività sono totalmente pratiche: all'inizio non vengono spiegati i concetti teorici che ne sono alla base, ma i ragazzi sono lasciati liberi di sperimentare.

Il facilitatore che guida ciascun gruppo ha proprio il compito di spingere gli studenti a formulare ipotesi e sperimentarle (proprio come si fa quando si programma e si fa "debug").

In una prima fase sarà proprio il facilitatore a impersonare Domo e a eseguire, con massima

precisione e senza compiere interpretazioni “umane”, le istruzioni che i bambini forniscono al robot. In una seconda fase egli potrà coinvolgere i bambini del gruppo per simulare l'esecuzione del programma (es. uno muoverà la pedina di Domo/si muoverà sulla scacchiera sul pavimento; un altro terrà in mano le carte e leggerà le istruzioni una per volta; nel caso di “blocchi” di carte annidati all'interno di una selezione o iterazione, manterrà il controllo della condizione dell'iterazione o della selezione, e passerà le carte al compagno affinché controlli l'esecuzione di quel blocco di carte, e così via).

Il facilitatore dovrà anche gestire le dinamiche interne ai gruppi: ad esempio, evitare che personalità dominanti prendano il sopravvento, assicurarsi che tutti partecipino alla discussione e comprendano le scelte fatte, anche in presenza di studenti particolarmente brillanti.

Alla fine della lezione, quando gli studenti hanno avuto esperienza pratica dei concetti introdotti, è possibile discutere insieme delle difficoltà, di ciò che si è imparato, e di formalizzare e consolidare la conoscenza.

Le attività sono altamente personalizzabili nell'ambientazione (e gli insegnanti sono invitati a calarle negli insegnamenti di altre discipline), nei tempi e nel livello di difficoltà.

Il percorso ha un'elevata sovrapposizione con traguardi e obiettivi della Proposta di indicazioni nazionali [CINI17] per le aree di algoritmi e di programmazione per la scuola primaria, tra cui:

- comprensione del concetto di algoritmo come rappresentazione di una procedura automatizzabile in modo preciso e non ambiguo (T-P-1)
- lettura, esecuzione mentale e correzione di errori in semplici programmi (T-P-3, T-P-4, O-P3-P-1, O-P5-P-1)
- scomposizione di un problema in parti più piccole (O-P3-A-2)
- (T-P-3) utilizzo di sequenza (O-P3-P-2), selezione a una (O-P3-P-4) o due vie (O-P5-P-5), iterazione per un numero fissato di volte (O-P3-P-3) o basata sulla verità di una condizione (O-P5-P-2)
- concetto di “blocco di istruzioni” come unica azione oggetto di ripetizione o selezione (O-P5-P-3)
- rappresentazione di informazioni e dati (le istruzioni del robot, in questo caso) tramite una convenzione arbitraria scelta dagli studenti (T-P-6, O-P3-D-2)

Attività con computer

Informatica creativa con Scratch

Questo percorso è stato svolto in una classe quarta di scuola primaria, dopo che, l'anno precedente, era stato sperimentato il percorso relativo a “Domo”.

Il percorso si basa su attività di Informatica Creativa, progettate nel contesto del cosiddetto “Creative Learning”⁸ (Apprendimento Creativo), framework pedagogico sviluppato al MIT di Boston e basato sulle teorie costruzioniste dell'apprendimento di S. Papert.

⁸ <https://learn.media.mit.edu/creative-learning>

Strumento principe per questo approccio è il linguaggio visuale a blocchi Scratch⁹.

Il percorso è stato realizzato in tre lezioni da tre ore ciascuna, ma è altamente personalizzabile tramite materiali disponibili in rete.

Gli studenti devono avere a disposizione dei computer (con connessione Internet e un browser che supporti Flash - questo requisito non sarà più richiesto da Scratch 3.0¹⁰).

I bambini possono lavorare da soli, se la scuola è attrezzata con un computer per ogni studente, o a coppie.

Il laboratorio è stato tenuto da un conduttore, che introduce i concetti e le sfide, con l'aiuto di uno o due altri facilitatori che - insieme al conduttore - girano tra i banchi supportando gli studenti nella realizzazione dei progetti.

Il conduttore ha creato un "Teacher account"¹¹ su Scratch e creato una classe, assegnando agli studenti username e password che non permettessero la loro identificazione (essendo gli account pubblici su Scratch - anche se limitatamente al tempo in cui si decide di tenere "aperta" la classe).

Gli argomenti delle lezioni sono dettagliati nel seguito.

Lezione 1

- Scratch surprise¹²: dopo una brevissima (2-3 minuti) introduzione all'interfaccia dell'editor di Scratch, viene lasciato agli studenti tempo per esplorare le potenzialità dello strumento, e si chiede a loro di far fare al gatto qualcosa di sorprendente.
- Login nella classe virtuale.
- Tutorial: l'insegnante svolge passo passo alla lavagna un progetto¹³, introducendo piano piano funzionalità o lasciando che gli studenti provino a realizzarle. La meccanica di gioco (e quindi il codice) sarà sostanzialmente uguale per tutti, ma gli studenti sono fortemente invitati a personalizzare l'ambientazione e ad aggiungere funzionalità una volta terminato il tutorial¹⁴. Questo primo progetto permette agli studenti di vedere una delle applicazioni di Scratch (spesso tra le più attraenti: la creazione di un videogioco). Il tutorial contiene però tutti gli elementi di base della programmazione strutturata (sequenza, selezione, iterazione, variabili, e anche eventi) - è ovviamente possibile semplificarlo (es. limitandosi a un personaggio che "mangia", uno che "viene mangiato" con relativo punteggio) a seconda del tempo e del livello della classe. In ogni caso può essere importante mostrare agli studenti altri esempi di progetti che

⁹ <https://scratch.mit.edu/>

¹⁰ Scratch 3.0 in versione beta è testabile a <https://beta.scratch.mit.edu/> e dovrebbe diventare la versione stabile a Gennaio 2019

¹¹ <https://scratch.mit.edu/educators/faq>

¹² <https://goo.gl/dAQyRR>

¹³ Ispirato a <https://goo.gl/he92PC>

¹⁴ Indicazioni su come realizzare il tutorial "In fondo al mar" (e altri tutorial) sono reperibili a <https://goo.gl/zr2U1N> e a <https://goo.gl/942rxD> grazie ad Angela Sofia Lombardo e all'associazione ProgrammaBOL

si possono realizzare con Scratch¹⁵ (es. storie interattive, arte) per evitare che associno lo strumento esclusivamente ai videogiochi.

- Salvataggio e condivisione: è importante spiegare agli studenti che sono incoraggiati a condividere i loro progetti, anche se incompleti, per ricevere feedback dagli altri utenti. Li si invita anche a scrivere la descrizione e le istruzioni del gioco, e a riconoscere eventuali crediti (es. idee, collaborazione col compagno, media presi dal web) nelle apposite aree nella pagina di condivisione. Questo punto è particolarmente importante per introdurre gli studenti all'idea di "codice sorgente aperto", alla base del software libero: tutti i progetti condivisi su Scratch infatti possono essere "guardati dentro" per studiare come sono stati realizzati e "remixati", cioè si può creare un nuovo progetto che fa uso di codice di altri (dandone in modo trasparente i crediti) - pratiche su cui si basa regolarmente la produzione di software professionale.
- Inserimento dei progetti in una "galleria di classe", che aiuta tutti a individuare ed esplorare i progetti degli altri.

Lezione 2

Durante la prima lezione molti studenti, invogliati dalla predisposizione naturale di Scratch ad essere esplorato liberamente, hanno manifestato la volontà di creare i propri progetti liberi.

Si è deciso pertanto di dedicare la seconda lezione alla creazione di progetti personali.

Agli studenti sono state date più di due ore per realizzare il progetto che preferivano. Per incentivarli ad avere un progetto "presentabile" alla fine del tempo, motivarli e permettere loro di ricevere feedback dai compagni, è stato anticipato loro che ognuno avrebbe proiettato, alla fine della lezione, la propria opera.

In situazioni di questo tipo, alcuni studenti avranno già idee per realizzare il proprio lavoro, mentre altri saranno spiazzati da tanta libertà.

Per questo agli studenti è stato suggerito di cercare aiuto su Scratch, e di esplorare altri progetti. Per facilitare gli studenti "bloccati", sono state stampate le cosiddette "cards", che danno idee o aiutano a implementare una specifica funzionalità (es. far cambiare colore a un personaggio).

Quelle utilizzate sono particolarmente snelle¹⁶ e più in linea con l'approccio creativo, mentre altre guidano passo passo nella realizzazione di un progetto iniziale¹⁷.

Il ruolo dei facilitatori è stato naturalmente quello di incoraggiare, di aiutare a risolvere specifici problemi (cercando però di non indirizzare gli studenti verso una specifica soluzione), ma anche quello di aiutare gli studenti a progettare il proprio artefatto di modo che fosse creativo ma anche fattibile per le loro potenzialità e per il tempo a disposizione.

Nel momento della presentazione è stato importante stimolare la riflessione, facendo domande sulle difficoltà incontrate, sulle cose che si sono imparate, su ciò che si vuole aggiungere, e stimolando gli altri bambini a fare commenti o critiche.

Lezione 3

Nella terza lezione si è cercato di utilizzare Scratch per insegnare alcuni concetti geometrici.

¹⁵ <https://scratch.mit.edu/tips>

¹⁶ <https://goo.gl/ffGX1J>

¹⁷ <https://goo.gl/WQifN3>

In particolare l'obiettivo finale era quello di realizzare un programma generale che, dato un qualsiasi $n > 2$, disegni il poligono regolare con n lati.

Una spiegazione dettagliata della lezione può essere trovata ad esempio in [GM16], un testo che utilizza l'apprendimento creativo e Scratch in diversi contesti disciplinari della scuola primaria.

Gli studenti dovrebbero dapprima sperimentare con la funzione "penna" di Scratch, che - in continuità con quanto svolto dalla tartaruga di LOGO, permette di "lasciare una traccia" al movimento degli oggetti, e dunque permette di disegnare.

Agli studenti viene poi chiesto di disegnare un quadrato, eventualmente mimando fisicamente una persona che cammina per un certo numero di passi, poi si gira di 90 gradi e così via per 4 volte.

Se gli studenti hanno ripetuto la coppia di istruzioni FAI PASSI e GIRA DI 90 gradi, è possibile far notare loro che è meglio utilizzare un blocco "ripeti 4 volte" - per esempio chiedendo loro cosa avrebbero fatto se gli fosse stato chiesto di disegnare un poligono con 20 lati, e facendoli riflettere sul fatto che inserendo una sola volta l'istruzione di movimento, si può cambiare il numero dei passi (e quindi la lunghezza del lato) cambiando una sola volta tale valore.

A questo punto agli studenti si chiede di disegnare un triangolo. L'errore tipico è quello di far girare lo sprite di 60 gradi (angolo interno) e non 120 gradi (angolo esterno).

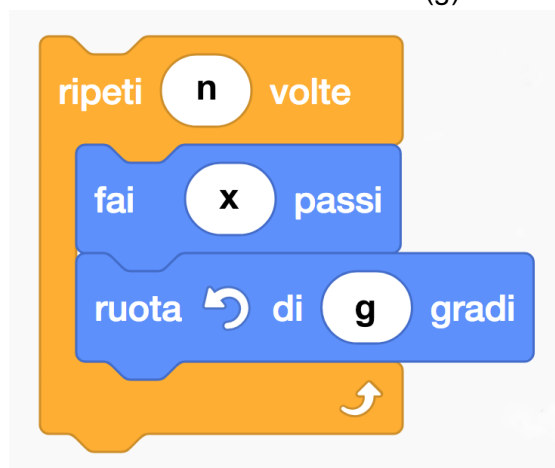
Ci si può aiutare ancora una volta con la teatralizzazione umana o con disegni alla lavagna.

A questo punto si chiede agli studenti di proseguire con le altre figure (pentagono, esagono...) ed essi dovrebbero notare che la struttura dei programmi è sempre la stessa:

RIPETI (n) VOLTE:

FAI (x) PASSI

RUOTA A SINISTRA DI (g) GRADI



dove (n) è il numero di lati del poligono, (x) è la lunghezza di un lato e (g) l'angolo esterno. Si può far osservare loro che tale g diminuisce all'aumentare dei lati:

lati (n)	gradi (g) (angolo esterno)
3	120
4	90
5	72

6	60
...	...

L'obiettivo è far osservare che un poligono regolare avrà tutti gli angoli esterni uguali, e che la loro somma deve per forza essere 360 perché il poligono "si chiuda" perfettamente.

In questo modo si cerca di insegnare costruttivamente una regola: *"la somma degli angoli esterni di un poligono è sempre 360"*, facendo sì che gli studenti abbiano sperimentato le conseguenze di tale regola *prima* che essa venga enunciata.

Durante le lezioni, gli studenti sono stati invitati a visitare il sito di Scratch anche a casa. Scratch è una community sicura per i bambini perché estremamente moderata dallo Scratch Team.

Inoltre, con un teacher account, il docente può controllare ed eliminare contenuti o commenti inappropriati.

I progetti sono comunque pubblici, e commentabili da tutta la comunità (ma non c'è modo di scambiarsi messaggi privati). I bambini sono stati quindi invitati a non dare informazioni personali e a segnalare agli insegnanti qualsiasi cosa sospetta.

Sono state inoltre insegnate alcune regole di base del comportamento sui social (es. evitare commenti offensivi). Molto importante è stata la lettura e la discussione delle "Linee guida della comunità di Scratch"¹⁸, regole di comportamento valide ben al di là della sola piattaforma. Questi aspetti sono stati discussi anche con i genitori che, inizialmente timorosi, hanno poi capito l'importanza di introdurre - tramite una community molto controllata ed educativa - i propri figli all'uso consapevole dei social network, ambiti dove si troveranno sicuramente ad interagire quando saranno più grandi.

Questa attività copre moltissimi punti della Proposta CINI [CINI17].

I traguardi più importanti che queste attività mirano a raggiungere sono ovviamente legati all'espressione di algoritmi mediante programmi in un certo linguaggio (T-P-2) e in generale alla scrittura di semplici programmi (T-P-3).

Alcune attività sono focalizzate a stimolare la libera espressione creativa dei bambini, e dunque tendono alla personalizzazione dell'apprendimento: studenti diversi potranno aver appreso, a livelli diversi, un range molto ampio di concetti diversi. A queste attività se ne possono far seguire altre più strutturate¹⁹, se si vuole che tutta la classe raggiunga alcuni specifici obiettivi negli ambiti di *algoritmi* e di *programmazione*.

Punto di forza di queste attività è, appunto, quello di far sperimentare ai bambini la possibilità di utilizzare applicazioni informatiche come strumento di espressione personale (T-P-11) e, nello specifico, di creare ed esprimersi tramite la realizzazione di programmi (storie, giochi, ...) (O-P5-R-2).

La discussione e l'attenzione alle norme di comportamento nella community di Scratch aiuta anche il raggiungimento di traguardi e obiettivi nell'ambito di uso consapevole delle nuove tecnologie, tra cui:

¹⁸ https://scratch.mit.edu/community_guidelines

¹⁹ Si vedano ad esempio i materiali <http://lia.disi.unibo.it/didatticaElementari/> realizzati in anni precedenti dallo stesso gruppo per la didattica nelle scuole da parte dei ricercatori del DISI - Università di Bologna

- regole per l'uso sicuro e responsabile delle tecnologie (T-P-9),
- riconoscimento di problematiche di privacy e riservatezza delle informazioni (O-P3-N-2),
- rispetto per gli altri (O-P3-N-3),
- comportamenti accettabili/inaccettabili con la tecnologia e con le informazioni da essa ottenute (O-P5-N-4),
- richiesta di aiuto da parte di adulti (O-P5-N-5).

Conclusioni

L'approccio sequenziale o intervallato di attività "unplugged" e "plugged" si è dimostrato efficace nelle esperienze maturate negli ultimi anni presso alcune scuole primarie di Bologna. Abbiamo potuto osservare come i bambini siano riusciti progressivamente a migliorare le capacità di ragionamento logico durante le attività unplugged proposte.

In particolare durante le ore dedicate a "Domo, il maggiordomo computazionale", abbiamo constatato che i bambini sono riusciti a risolvere con crescente padronanza problemi di complessità sempre maggiore. Particolarmente entusiasmante per i bambini è risultata la possibilità di testare fisicamente le loro soluzioni, simulando a turno l'esecuzione del programma.

Quello che si è potuto osservare è stata poi la capacità di applicare il metodo risolutivo - appreso tramite le attività unplugged proposte - alla risoluzione di problemi tramite Scratch l'anno successivo. Le attività unplugged hanno permesso ai bambini di interiorizzare alcuni concetti fondamentali per la formulazione di algoritmi, quali l'utilizzo di sequenza, selezione a una o due vie, iterazione per un numero fissato di volte (e anche iterazione basata sulla verità di una condizione) con condizioni semplici. Questa interiorizzazione ha permesso ai bambini di orientarsi rapidamente con l'ambiente Scratch, individuando e utilizzando facilmente in esso gli schemi di controllo di flusso per risolvere le sfide proposte o per realizzare giochi, storie e animazioni. L'efficacia osservata dai docenti universitari avrebbe poco senso se non fosse stata in qualche modo confermata dall'insegnante che ha accompagnato i bambini nelle attività proposte. Ella ha in particolare osservato la loro disponibilità dei bambini a lavorare, soprattutto in attività nuove rispetto al contesto didattico usuale, contesto nel quale tendono a lavorare singolarmente perseguendo un proprio obiettivo.

Inizialmente l'argomento è parso ostico all'insegnante stessa, ma ben presto essa ha constatato l'assoluta disponibilità dei suoi studenti a nuove esperienze. Molto apprezzato dalla docente anche il fatto che l'attività sia stata presentata senza sovrastrutture "tecniche" che avrebbero reso più difficile la comprensione e forse avrebbero spaventato i bambini. La maestra ha inoltre notato che in un contesto così diverso dall'usuale, alcuni dei bambini apparentemente meno sicuri di sé hanno saputo dare risposte e trovare soluzioni per il proprio gruppo, traendone grande soddisfazione.

Il fatto di avere svolto attività pratiche unplugged - da riprodurre in classe o in palestra coinvolgendo il proprio corpo nella simulazione dell'esecuzione di algoritmi - ha giovato molto ed ha facilitato i bambini, permettendo loro di comprendere la necessità di dare istruzioni di un certo tipo alle macchine. Tutti gli adulti coinvolti (maestra e docenti del DISI) hanno notato che, a distanza di quasi un anno, tutti i gruppi hanno saputo riutilizzare nelle attività con Scratch ciò che avevano messo in pratica l'anno precedente nelle attività unplugged.

Riferimenti bibliografici

[BV18] Tim Bell, Jan Vahrenhold (2018). CS Unplugged—How Is It Used, and Does It Work?. In: Böckenhauer HJ., Komm D., Unger W. (eds) Adventures Between Lower Bounds and Higher Altitudes. Lecture Notes in Computer Science, vol 11011. Springer, Cham

[CINI17] CINI (2017). Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola <https://www.consortio-cini.it/images/Proposta-Indicazioni-Nazionali-Informatica-Scuola-numerata.pdf>

[DLM17a] Renzo Davoli, Michael Lodi, Rebecca Montanari (2017). Domo: un maggiordomo “computazionale”. <http://www.cs.unibo.it/~renzo/DIDATTICA/DomoCompleto.pdf>

[DLM17b] Renzo Davoli, Michael Lodi, Rebecca Montanari (2017). Domo: un maggiordomo “computazionale” - Materiali stampabili. <http://www.cs.unibo.it/~renzo/DIDATTICA/domo>

[GM16] Marco Giordano, Caterina Moschetti (2016). Coding e pensiero computazionale nella scuola primaria. ELI - La Spiga edizioni.